 Ruben Verborgh

Piecing the puzzle

Self-publishing queryable research data on the Web

Ruben Verborgh, Ghent University – imec – IDLab *20 January 2017*

Publishing research on the Web accompanied by machine-readable data is one of the aims of Linked Research. Merely embedding metadata as RDFa in HTML research articles, however, does not solve the problems of accessing and querying that data. Hence, I created a simple ETL pipeline to extract and enrich Linked Data from my personal website, publishing the result in a queryable way through Triple Pattern Fragments. The pipeline is open source, uses existing ontologies, and can be adapted to other websites. In this article, I discuss this pipeline, the resulting data, and its possibilities for query evaluation on the Web. More than 35,000 RDF triples of my data are queryable, even with federated SPARQL queries because of links to external datasets. This proves that researchers do not need to depend on centralized repositories for readily accessible (meta-)data, but instead can—and should—take matters into their own hands.

Introduction

The World Wide Web continues to shape many domains, and not in the least research. On the one hand, the Web beautifully fulfills its role as a *distribution channel* of scientific knowledge, for which it was originally invented. This spurs interesting dialogues concerning **Open Access** [1] and even **piracy** [2] of research articles. On the other hand, the advent of *social networking* creates new interaction opportunities for researchers, but also forces us to consider our **online presence** [3]. Various social networks dedicated to research have emerged: **Mendeley**, **ResearchGate**, **Academia**, … They attract millions of researchers, and employ various **tactics** to keep us there.

A major issue of these social research networks is their *lack of mutual complementarity*. None of them has become a clear winner in terms of adaption. At first sight, the resulting plurality seems a blessing for diversity, compared to the monoculture of Facebook for social networking in general. Yet whereas other generic social networks such as Twitter and LinkedIn serve complementary professional purposes compared to Facebook, social *research* networks share nearly identical goals. As an example, a researcher could announce a newly accepted paper on Twitter, discuss its review process on Facebook, and share a photograph of an award on LinkedIn. In contrast, one would typically not exclusively list a specific publication on Mendeley and another on Academia, as neither publication list would be complete.

In practice, this results in constant bookkeeping for researchers who want each of their profiles to correctly represent them—a necessity if such profiles are implicitly or explicitly treated as **performance indicators** [4]. Deliberate absence on any of these networks is not a viable option, as parts of one's publication metadata might be automatically harvested or entered by co-authors, leaving an automatically generated but incomplete profile. Furthermore, the quality of such non-curated metadata records can be questionable. As a result, researchers who do not actively maintain their online research profiles risk ending up with *incomplete* and *inaccurate* publication lists on those networks. Such misrepresentation can be significantly worse than not being present at all—but given the public nature of publication metadata, complete absence is not an enforceable choice.

Online representation is not limited to social networks: scientific publishers also make metadata available about their journals and books. For instance, Springer Nature recently released **SciGraph**, a Linked Open Data platform that includes scholarly metadata. *Accuracy* is less of an issue in such cases, as data comes directly from the source. However, *quality* and *usability* are still influenced by the way data is modeled and whether or how identifiers are disambiguated. *Completeness* is not guaranteed, given that authors typically target multiple publishers. Therefore, even such authoritative sources do not provide individual researchers with a correct profile.

In the spirit of **decentralized social networking** [5] and **Linked Data** [6], several researchers instead started publishing their own data and metadata. I am one of them, since I believe in **practicing what we preach** [7] as Linked Data advocates, and because I want my own website to act as the main authority for my data. After all, I can spend more effort on the completeness and accuracy of my publication metadata than most other platforms could reasonably do for me. In general, self-published data typically resides in separate **RDF documents** [8] (for which the **FOAF vocabulary** [9] is particularly **popular** [10]), or inside of HTML documents (using **RDFa Lite** [11] or similar formats).

Despite the controllable quality of personally maintained research data and metadata in individual documents on the Web, they are not as *visible*, *findable*, and *queryable* as those of social research networks. I call a dataset interface "queryable" with respect to a given query when a consumer does not need to download the entire dataset in order to evaluate that query over it with full completeness. Unfortunately, hosting advanced search interfaces on a personal website quickly becomes complex and expensive. To mitigate this, I have implemented a simple *Extract/Transform/Load (ETL) pipeline* on top of my personal website, which extracts, enriches, and publishes my Linked Data in a queryable way through a **Triple Pattern Fragments** [12] interface. The resulting data can be **browsed** and **queried** live on the Web, with higher quality and flexibility than on my other online profiles, and at only a limited cost for me as data publisher.

This article describes my **use case**, which resembles that of many other researchers. I detail the design and implementation of the **ETL pipeline**, and report on its **results**. At the end, I list **open questions** regarding self-publication, before **concluding** with a reflection on the opportunities for the broader research community.

Use case

Available data

Like the websites of many researchers, my **personal website** contains data about the following types of resources:

- **people** such as colleagues, collaborators, and fellow researchers
- **research articles** I have co-authored
- **blog posts** I have written
- **courses** I teach

This data is spread across different HTTP resources:

- a single **RDF document (FOAF profile)** containing:
  - manually entered data *(personal data, affiliations, projects, …)*
  - automatically generated metadata *(publications, blog posts, …)*
- an **HTML page with RDFa** per:
  - publication *(publication and author metadata)*
  - blog post *(post metadata)*
  - HTML article *(metadata and citations)*
  - …

Depending on the context, I encode the information with different vocabularies:

- **Friend of a Friend (FOAF)** *(people, documents, …)*
- **Schema.org** *(blog posts, articles, courses, …)*
- **Bibliographic Ontology (BIBO)** *(publications)*
- **Citation Typing Ontology (CiTO)** *(citations)*
- …

There is a considerable amount of *overlap* since much data is available in more than one place, sometimes in different vocabularies. For example, webpages about my publications contain Schema.org markup (to facilitate indexing by search engines), whereas my profile describes the same publications more rigorously using BIBO and FOAF (for more advanced RDF clients). I deliberately reuse the same identifiers for the same resources everywhere, so identification is not an issue.

Data publication requirements

While the publication of structured data as RDF and RDFa is conveniently integrated in the webpage creation process, *querying information over the entire website* is difficult. For instance, starting from the homepage, obtaining a list of all mentioned people on the website would be non-trivial. In general, SPARQL query execution over Linked Data takes a considerable amount of time, and **completeness cannot be guaranteed** [13]. So while Linked Data documents are excellent for automated exploration of individual resources, and for aggregators such as search engines that can harvest the entire website, the possibilities of individual automated clients remain limited.

Another problem is the *heterogeneity of vocabularies*: clients without reasoning capabilities would only find subsets of the information, depending on which vocabulary is present in a given representation. Especially in RDFa, it would be cumbersome to combine every single occurrence of `schema:name` with the semantically equivalent `dc:title`, `rdfs:label`, and `foaf:name`. As such, people might have a `foaf:name` (because FOAF is common for people), publications a `schema:name` (because of `schema:ScholarlyArticle`), and neither an `rdfs:label`. Depending on the kind of information, queries would thus need different predicates for the concept “label”. Similarly, queries for `schema:Article` or `schema:CreativeWork` would not return results because they are not explicitly mentioned, even though their subclasses `schema:BlogPosting` and `schema:ScholarlyArticle` appear frequently.

Given the above considerations, the constraints of individual researchers, and the possibilities of social research networks, we formulate the following requirements:

- Automated clients should be able to evaluate **queries with full completeness** with respect to the data on the website.
- **Semantically equivalent expressions** should yield the same query results, **regardless of vocabulary** with respect to all vocabularies used on the website.
- Queryable data can only involve a **limited cost and effort** for publishers as well as consumers.

ETL pipeline

To automate this process, I have developed a simple ETL pipeline. With the exception of a couple of finer points, the pipeline itself is fairly straightforward. What is surprising, however, is the impact such a simple pipeline can have, as discussed hereafter in the **Results** section. The pipeline consists of the following phases, which will be discussed in the following subsections.

- **Extract** all triples from the website’s RDF and HTML+RDFa documents.
- **Reason** over this data and its ontologies to complete gaps.
- **Publish** the resulting data in a queryable interface.

The **source code** for the pipeline is available on GitHub. The pipeline can be run periodically, or triggered on website updates as part of a continuous integration process. In order to adapt this to different websites, the **default ontology files** can be replaced by others that are relevant for a given website.

Extract

The pipeline loops through all of the website's files (either through the local file system or through Web crawling) and makes lists of RDF documents and HTML+RDFa documents. The RDF documents are fed through the **Serd parser** to verify validity and for conversion into **N-Triples** [14], so the rest of the pipeline can assume one triple per line. The RDFa is parsed into N-Triples by the **RDFLib library** for Python. Surprisingly, this library was the only one I found that correctly parsed RDFa Lite in (valid) HTML5; both **Raptor** and **Apache Any23** seemed to expect a stricter document layout.

Reason

In order to fix gaps caused by implicit properties and classes, the pipeline performs reasoning over the extracted data and its ontologies to compute the deductive closure. The choice of ontologies is based on the data, and currently includes **FOAF**, **DBpedia**, **CiTO**, **Schema.org**, and the **Organizations ontology**. Additionally, I specified a limited number of **custom OWL triples** to indicate equivalences that hold on my website, but not necessarily in other contexts.

The pipeline delegates reasoning to the highly performant **EYE reasoner** [15], which does not have any RDFS or OWL knowledge built-in. Consequently, relevant **RDFS and OWL theories** can be selected manually, such that only a practical subset of the entire deductive closure is computed. For instance, my FOAF profile asserts that all resources on my site are different using `owl:AllDifferent`; a full deductive closure would result in an undesired combinatorial explosion of `owl:differentFrom` statements.

The website's dataset is enriched through the following steps:

1. The ontologies are **skolemized** [8] and concatenated into a single ontology file.
2. The **deductive closure of the joined ontology** is computed by passing it to the EYE reasoner with the RDFS and OWL theories.
3. The **deductive closure of the website's data** is computed by passing it to the EYE reasoner with the RDFS and OWL theories and the deductive closure of the ontology.
4. **Ontological triples are removed from the data** by subtracting triples that also occur in the deductive closure of the ontology.
5. **Other unnecessary triples are removed**, in particular triples with skolemized ontology IRIs, which are meaningless without the ontology.

These steps ensure that only triples directly related to the data are published without any direct or derived triples from its ontologies, which form different datasets. By separating them, ontologies remain published as independent datasets, and users executing queries can explicitly choose which ontologies or datasets to include.

For example, when the original data contains

```
art:publication schema:author rv:me.
```

and given that DBpedia and Schema.org ontologies (before skolemization) contain

```
dbo:author owl:equivalentProperty schema:author.
schema:author rdfs:range [
  owl:unionOf (schema:Organization schema:Person)
].
```

then the raw reasoner output of step 3 (after skolemization) would be

```
6  art:publication dbo:author rv:me.
7  art:publication schema:author rv:me.
8  rv:me rdf:type skolem:b0.
9  dbo:author owl:equivalentProperty schema:author.
10 schema:author rdfs:range skolem:b0.
11 skolem:b0 owl:unionOf skolem:l1.
12 skolem:l1 a rdf:List.
13 skolem:l1 rdf:first schema:Organization.
14 skolem:l1 rdf:rest  skolem:l2.
15 skolem:l2 a rdf:List.
16 skolem:l2 rdf:first schema:Person.
17 skolem:l2 rdf:rest  rdf:nil.
```

The skolemization in step 1 ensures that blank nodes from ontologies have the same identifier before and after the reasoning runs in steps 2 and 3. Step 2 results in triples 9–17 (note the inferred triples 12 and 15), which are also present in the output of step 3, together with the added triples 6–8 derived from data triple 1. Because of the previous skolemization, triples 9–16 can be removed through a simple line-by-line difference, as they have identical N-Triples representations in the outputs of steps 2 and 3. Finally, step 5 removes triple 8, which is not meaningful as it points to an unreferenceable blank node in the Schema.org ontology. The resulting enriched data is:

```
18 art:publication dbo:author rv:me.
19 art:publication schema:author rv:me.
```

Thereby, data that was previously only described with Schema.org in RDFa becomes also available with DBpedia. Note that the example triple yields several more triples in the actual pipeline, which uses the full FOAF, Schema.org, and DBpedia ontologies.

Passing the deductive closure of the joined ontology from step 2 to step 3 improves performance, as the derived ontology triples are already materialized. Given that ontologies change slowly, the output of steps 1 and 2 could be cached.

Publish

The resulting triples are then published through a **Triple Pattern Fragments (TPF)** [12] interface, which allows clients to access a dataset by triple pattern. In essence, the lightweight TPF interface extends Linked Data's subject-based dereferencing by also providing predicate- and object-based lookup. Through this interface, clients can execute SPARQL queries with full completeness at limited server cost. Because of the simplicity of the interface, various back-ends are possible. For instance, the data from the pipeline can be served from memory by loading the generated N-Triples file, or the pipeline can compress it into a **Header Dictionary Triples (HDT)** [16] file.

Special care is taken to make IRIs **dereferenceable** [6] during the publication process. While I emphasize IRI reuse, some of my co-authors do not have their own profile, so I had to mint IRIs for them. Resolving such IRIs results in an HTTP 303 redirect to the TPF with data about the concept. For instance, the IRI `https://data.verborgh.org/people/sam_coppens` redirects to the TPF of **triples with this IRI as subject**.

Results

I applied the ETL pipeline to my personal website **https://ruben.verborgh.org/** to verify its effectiveness. The data is published at **https://data.verborgh.org/ruben** and can be queried with a TPF client such as

http://query.verborgh.org/. The results reflect the status of January 2017, and measurements were executed on a MacBook Pro with a 2.66GHz Intel Core i7 processor and 8GB of RAM.

Generated triples

In total, 35,916 triples were generated in under 5 minutes from 6,307 profile triples and 12,564 unique triples from webpages. The table below shows the number of unique triples at each step and the time it took to obtain them. The main bottleneck is *not* reasoning (≈3,000 triples per second), but rather RDFa extraction (≈100 triples per second), which can fortunately be parallelized more easily.

step	time (s)	# triples
RDF(a) extraction	170.0	17,050
ontology skolemization	0.6	44,179
deductive closure ontologies	38.8	144,549
deductive closure data and ontologies	61.8	183,282
subtract ontological triples	0.9	38,745
subtract other triples	1.0	35,916
total	**273.0**	**35,916**

Table 1: The number of unique triples per phase, and the time it took to extract them.

While **dataset size is not an indicator for quality** [17], the accessibility of the data improves through the completion of inverse predicates and equivalent or subordinate predicates and classes between ontologies. The table below lists the frequency of triples with specific predicates and classes before and after executing the pipeline.

predicate or class	# pre	# post
`dc:title`	657	714
`rdfs:label`	473	714
`foaf:name`	394	714
`schema:name`	439	714
`schema:isPartOf`	263	263
`schema:hasPart`	0	263
`cito:citesAsAuthority`	14	14
`cito:cites`	0	33
`schema:citation`	0	33
`foaf:Person`	196	196
`dbo:Person`	0	196
`schema:ScholarlyArticle`	203	203
`schema:Article`	0	243
`schema:CreativeWork`	0	478

Table 2: The number of triples with the given predicate or class before and after the execution of the pipeline, grouped by semantic relatedness.

It is important to note that most improvements are solely the result of reasoning on **existing ontologies**; only **8 custom OWL triples** were added (7 for equivalent properties, 1 for a symmetric property).

Quality

While computing the deductive closure should not introduce any inconsistencies, the quality of the ontologies directly impacts the result. While inspecting the initial output, I found the following conflicting triples, typing me as a person *and* a company:

```
rv:me rdf:type dbo:Person.
rv:me rdf:type dbo:Company.
```

To find the cause of this inconsistency, I ran the reasoner on the website data and ontologies, but instead of asking for the deductive closure, I asked to prove the second triple. The resulting proof traced the result back to the DBpedia ontology erroneously stating the equivalence of the `schema:publisher` and `dbo:firstPublisher` properties. While the former has both people and organisations in its range, the latter is specific to companies—hence the conflicting triple in the output. I reported this **issue** and manually corrected it in the ontology. Similarly, `dbo:Website` was **deemed equivalent** to `schema:WebPage`, whereas the latter should be `schema:WebSite`. Disjointness constraints in the ontologies would help catch these mistakes. Further validation with **RDFUnit** [18] brought up a list of errors, but all of them turned out to be false positives.

Queries

Finally, I report on the execution time and number of results for a couple of example SPARQL queries. These were evaluated against the **live TPF interface** by a TPF client, and against the actual webpages and profile by a Linked Data-traversal-based client (**SQUIN** [19]). The intention is *not* to compare these query engines, as they use different paradigms and query semantics: TPF guarantees 100% completeness with respect to given datasets, whereas SQUIN considers reachable subwebs. The goal is rather to highlight the limits of querying over RDFa pages as practiced today, and to contrast this with the improved dataset resulting from the ETL pipeline.

To this end, I tested three scenarios on the public Web:

1. a Triple Pattern Fragments client (**ldf-client 2.0.4**) with the pipeline's **TPF interface**
2. a Linked Data client (**SQUIN 20141016**) with my **homepage** as seed
3. a Linked Data client (**SQUIN 20141016**) with my **FOAF profile** as seed

All clients started with an empty cache for every query, and the query timeout was set to 60 seconds. The waiting period between requests for SQUIN was disabled. For the federated query, the TPF client also accessed **DBpedia**, which the Linked Data client can find through link traversal. To highlight the impact of the seeds, queries avoid IRIs from my domain by using literals for concepts instead.

	TPF *(pipeline)*		LD *(home)*		LD *(profile)*	
query	**#**	**t** (s)	**#**	**t** (s)	**#**	**t** (s)
people I know (`foaf:name`)	196	2.1	0	5.6	14	60.0
people I know (`rdfs:label`)	196	2.1	0	3.2	200	60.0
publications I wrote	205	4.0	0	10.8	0	10.5
my publications	205	4.1	134	12.6	134	14.4
my blog posts	43	1.1	40	6.5	40	6.4
my articles	248	4.9	0	6.3	0	3.3
a colleague's publications	32	1.1	20	13.9	20	16.3
my first-author publications	46	2.7	0	3.8	6	36.2
works I cite	33	0.5	0	4.0	0	60.0
my interests *(federated)*	4	0.4	0	4.0	4	1.8

Table 3: Number of results and execution time per query, comparing the TPF client on the enhanced data with Linked Data traversal on my website (starting from my home page or my FOAF profile).

The first two queries show the influence of *ontological equivalences*. At the time of writing, my website related me to 196 `foaf:Persons` through the `foaf:knows` predicate. If the query uses only the FOAF vocabulary, with `foaf:name` to obtain people's names, Linked Data traversal finds 14 results. If we use

`rdfs:label` instead, it even finds additional results on external websites (because of link-traversal query semantics).

A second group of queries reveals the impact of *link unidirectionality* and inference of *subclasses and subproperties* in queries for scholarly publications and blog posts. Through traversal, *"publications I wrote"* (with `foaf:made`) does not yield any results, whereas *"my publications"* (with `schema:author`) yields 134, even though both queries are semantically equivalent. Given that my profile actually contained 205 publications, the 71 missing publications are caused by SQUIN's implementation rather than being an inherent Linked Data limitation. Blog posts are found in all scenarios, even though the traversal client finds 3 fewer posts. Only the TPF client is able to find all articles, because the pipeline generated the inferred type `schema:Article` for publications and blog posts. Other more constrained queries for publications yield fewer results through traversal as well. Citations (`cito:cites`) are only identified by the TPF client, as articles solely mention its subproperties.

The final test examines a *federated query:* when starting from the profile, the Linked Data client also finds all results.

Regarding execution times, the measurements provide positive signals for low-cost infrastructures on the public Web. Note that both clients return results *iteratively*. With an average arrival rate of 53 results per second for the above queries, the TPF client's pace exceeds the processing capabilities of people, enabling usage in live applications. Even faster performance could be reached with, for instance, a data dump or a SPARQL endpoint; however, these would involve an added cost for either the data publisher or consumer, and might have difficulties in federated contexts.

Open questions

Publishing RDFa data on my website over the past years—and subsequently creating the above pipeline—has left me with a couple of questions, some of which I discuss below.

A first question is *what* data should be encoded as Linked Data, and how it should be *distributed* across resources. In the past, I always had to decide whether to write data directly on the page as HTML+RDFa, whether to place it in my FOAF profile as RDF, whether to do both, or neither. The pipeline partially solves the *where* problem by gathering all data in a single interface. Even though each page explicitly links to the Linked Data-compatible TPF interface using `void:inDataset`—so traversal-based clients can also consume it—other clients might only extract the triples from an individual page. Furthermore, apart from the notable exception of search engine crawlers, it is hard to predict what data automated clients are looking for.

A closely related question is *what ontologies* should be used on which places. Given that authors have limited time and in order to not make HTML pages too heavy, we should probably limit ourselves to a handful of vocabularies. When inter-vocabulary links are present, the pipeline can then materialize equivalent triples automatically. I have chosen Schema.org for most HTML pages, as this is consumed by several search engines. However, this vocabulary is rather loose and might not fit other clients. Perhaps the FOAF profile is the right place to elaborate, as this is a dedicated RDF document that attracts more specific-purpose clients compared to regular HTML pages.

Even after the above choices have been made, the *flexibility* of some vocabularies leads to additional decisions. For example, in HTML articles I mark up citations with the **CiTO ontology**. The domain and range of predicates such as `cito:cites` is open to documents, sections, paragraphs, and other units of information. However, choosing to cite an article from a paragraph influences how queries such as "citations in my articles" need to be written. Fortunately, the pipeline can infer the other triples, such that the section and document containing the paragraph also cite the article.

When marking up data, I noticed that I sometimes attach *stronger meaning* to concepts than strictly prescribed by their ontologies. Some of these semantics are encoded in my **custom OWL triples**, whose contents contribute to the reasoning process (but do not appear directly in the output, as this would leak my semantics globally). For instance, I assume equivalence of `rdfs:label` and `foaf:name` for my purposes, and treat the `foaf:knows` relation as symmetrical (as in its textual—but not formal—definition). Using my own subproperties in these cases would encode more specific semantics, while the other properties could be derived from the pipeline. However, this would require maintaining a custom ontology, to which few queries would refer.

The *reuse of identifiers* is another source of debate. I opted as much as possible to reuse URLs for people and publications. The advantage is that this enables Linked Data traversal, so additional RDF triples can be picked up from FOAF profiles and other sources. The main drawback, however, is that the URLs do not dereference to my own datasource, which *also* contains data about their concepts. As a result, my RDF data contains a mix of URLs that dereference externally (such as **http://csarven.ca/#i**), URLs that dereference to my website (such as **https://ruben.verborgh.org/articles/queryable-research-data/**) and URLs that dereference to my TPF interface (such as **https://data.verborgh.org/people/anastasia_dimou**) Fortunately, the TPF interface can be considered an **extension of the Linked Data principles** [20], such that URLs can be "dereferenced" (or queried) on different domains as well, yet this not help regular Linked Data crawlers. An alternative is using my own URLs everywhere and connecting them with external URLs through `owl:sameAs`, but then certain results would only be revealed to more complex SPARQL queries that explicitly consider multiple identifiers.

With regard to publishing, I wondered to what extent we should place RDF triples in the *default graph* on the Web at large. As noted above, inconsistencies can creep in the data; also, some of the things I state might reflect my beliefs rather than general truths. While RDFa does not have a standardized option to place data in named graphs, other types of RDF documents do. By moving my data to a dedicated graph, as is practiced by several datasets, I could create a separate context for these triples. This would also facilitate provenance and other applications, and it would then be up to the data consumer to decide how to treat data graph.

The above questions highlight the need for *guidance* and *examples* in addition to specifications and standards. *Usage statistics* could act as an additional information source. While HTTP logs from the TPF interface do not contain full SPARQL queries, they show the IRIs and triple patterns clients look for. Such behavioral information would not be available from clients or crawlers visiting HTML+RDFa pages.

Finally, when researchers start self-publishing their data in a queryable way at a large scale, we will need a *connecting layer* to approach the decentralized ecosystem efficiently through a single user interface. While federated query execution over multiple TPF interfaces on the public Web is feasible, as demonstrated above, this mechanism is impractical to query hundreds or thousands of such interfaces. On the one hand, this indicates their will still be room for centralized indexes or aggregators, but their added value then shifts from data to services. On the other hand, research into decentralized technologies might make even such indexes obsolete.

Conclusion

RDFa makes semantic data *publication* easy for researchers who want to be in control of their online data and metadata. For those who prefer not to work directly on RDFa, or lack the knowledge to do so, annotation tools and editors can help with its *production*. In this article, I examined the question of how we subsequently can optimize the *queryability* of researchers' data on the Web, in order to facilitate their *consumption* by different kinds of clients.

Simple clients do not possess the capabilities of large-scale aggregators to obtain all Linked Data on a website. They encounter mostly individual HTML+RDFa webpages, which are always incomplete with respect to both the whole of knowledge on a website as well as the ontological constructs to express it. Furthermore, variations in reasoning capabilities make bridging between different ontologies difficult. The proposed ETL pipeline addresses these challenges by publishing a website's explicit and inferred triples in a queryable interface. The pipeline itself is simple and can be ported to different scenarios. If cost is an issue, the extraction and reasoning steps can run on public infrastructures such as **Travis CI**, as all involved software is open source. Queryable data need not be expensive either, as proven by **free TPF interfaces on GitHub** [21] and by the **LOD Laundromat** [22], which provides more than 600,000 TPF interfaces on a single server.

By publishing queryable research data, we contribute to the **Linked Research** vision: the proposed pipeline increases reusability and improves linking by completing semantic data through reasoning. The possibility to execute live queries—and in particular federated queries—enables new use cases, offering researchers additional incentives to self-publish their data. Even though I have focused on research data, the principles generalize to other domains. In particular, the **Solid** project for decentralized social applications could benefit from a similar pipeline to facilitate data querying and exchange across different parties in a scalable way.

Even as a researcher who has been publishing RDFa for years, I have often wondered about the significance of adding markup to individual pages. I doubted to what extent the individual pieces of data I created contributed to the larger puzzle of Linked Data on my site and other websites like it, given that they only existed within the confines of a single page. Building the pipeline enabled the execution of complex queries across pages, without significantly changing the maintenance cost of my website. From now on, every piece of data I mark up directly leads to one or more queryable triples, which provides me with a stronger motivation. If others follow the same path, we no longer need centralized data stores. We could execute federated across researchers' websites, using combinations of Linked Data traversal and more complex query interfaces that can guarantee completeness. Centralized systems can play a crucial role by providing indexing and additional services, yet they should act at most as secondary storage.

Unfortunately, exposing my own data in a queryable way does not relieve me yet of my frustration of synchronizing that data on current social research networks. It does make my data more searchable and useful though, and I deeply hope that one day, these networks will synchronize with my interface instead of the other way round. Most of all, I hope that others will mark up their webpages and make them queryable as well, so we can query research data on the *Web* instead of in silos. To realize this, we should each contribute our own pieces of data in a way that makes them fit together easily, instead of watching third parties mash our data into an entirely different puzzle altogether.

References

[1] Harnad, S. and Brody, T. (2004), "Comparing the Impact of Open Access (OA) vs. Non-OA Articles in the Same Journals", *D-Lib Magazine*, June, available at: http://www.dlib.org/dlib/june04/harnad/06harnad.html.

[2] Bohannon, J. (2016), "Who's downloading pirated papers? Everyone", *Science*, American Association for the Advancement of Science, Vol. 352 No. 6285, pp. 508–512, available at: http://science.sciencemag.org/content/352/6285/508.

[3] Van Noorden, R. (2014), "Online collaboration: Scientists and the social network", *Nature*, Vol. 512 No. 7513, pp. 126–129, available at: http://www.nature.com/news/online-collaboration-scientists-and-the-social-network-1.15711.

[4] Thelwall, M. and Kousha, K. (2015), "Web indicators for research evaluation: Part 2: Social media metrics", *El Profesional De La Información*, EPI SCP, Vol. 24 No. 5, pp. 607–620, available at: http://www.elprofesionaldelainformacion.com/contenidos/2015/sep/09.pdf.

[5] Yeung, C.-man A., Liccardi, I., Lu, K., Seneviratne, O. and Berners-Lee, T. (2009), "Decentralization: The future of online social networking", in *Proceedings of the W3C Workshop on the Future of Social Networking Position Papers*, Vol. 2, pp. 2–7, available at: https://www.w3.org/2008/09/msnws/papers/decentralization.pdf.

[6] Berners-Lee, T. (2006), "Linked Data", July, available at: https://www.w3.org/DesignIssues/LinkedData.html.

[7] Möller, K., Heath, T., Handschuh, S. and Domingue, J. (2007), "Recipes for Semantic Web Dog Food – The ESWC and ISWC Metadata Projects", in Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., et al. (Eds.), *Proceedings of 6th International Semantic Web Conference*, Vol. 4825, Lecture Notes in Computer Science, pp. 802–815, available at: http://iswc2007.semanticweb.org/papers/795.pdf.

[8] Cyganiak, R., Wood, D. and Lanthaler, M. (Eds.). (2014), *RDF 1.1 Concepts and Abstract Syntax*, Recommendation, World Wide Web Consortium, available at: https://www.w3.org/TR/rdf11-concepts/.

[9] Brickley, D. and Miller, L. (2014), "FOAF Vocabulary Specification 0.99", available at: http://xmlns.com/foaf/spec/.

[10] Ding, L., Zhou, L., Finin, T. and Joshi, A. (2005), "How the Semantic Web is Being Used: An Analysis of FOAF Documents", in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, available at: http://ebiquity.umbc.edu/_file_directory_/papers/120.pdf.

[11] Sporny, M. (Ed.). (2015), *RDFa Lite 1.1 – Second Edition*, Recommendation, World Wide Web Consortium, available at: https://www.w3.org/TR/rdfa-lite/.

[12] Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., Haesendonck, G., et al. (2016), "Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web", *Journal of Web Semantics*, Vol. 37–38, pp. 184–206, available at: http://linkeddatafragments.org/publications/jws2016.pdf.

[13] Hartig, O. (2013), "An Overview on Execution Strategies for Linked Data Queries", *Datenbank-Spektrum*, Springer, Vol. 13 No. 2, pp. 89–99, available at: http://olafhartig.de/files/Hartig_LDQueryExec_DBSpektrum2013_Preprint.pdf.

[14] Beckett, D. (2014), *RDF 1.1 N-Triples*, Recommendation, World Wide Web Consortium, available at: https://www.w3.org/TR/n-triples/.

[15] Verborgh, R. and De Roo, J. (2015), "Drawing Conclusions from Linked Data on the Web", *IEEE Software*, Vol. 32 No. 5, pp. 23–27, available at: http://online.qmags.com/ISWO515?cid=3244717&eid=19361&pg=25.

[16] Fernández, J.D., Martínez-Prieto, M.A., Gutiérrez, C., Polleres, A. and Arias, M. (2013), "Binary RDF Representation for Publication and Exchange (HDT)", *Journal of Web Semantics*, Elsevier, Vol. 19, pp. 22–41, available at: http://www.websemanticsjournal.org/index.php/ps/article/view/328.

[17] Vrandecíc Denny, Krötzsch, M., Rudolph, S. and Lösch, U. (2010), "Leveraging non-lexical knowledge for the linked open data web", *Review of April Fool's Day Transactions*, Vol. 5, pp. 18–27, available at: http://km.aifb.kit.edu/projects/numbers/linked_open_numbers.pdf.

[18] Kontokostas, D., Westphal, P., Auer, S., Hellmann, S., Lehmann, J., Cornelissen, R. and Zaveri, A. (2014), "Test-driven Evaluation of Linked Data Quality", in *Proceedings of the 23rd International Conference on World Wide Web*, ACM, pp. 747–758, available at: http://svn.aksw.org/papers/2014/WWW_Databugger/public.pdf.

[19] Hartig, O. (2011), "Zero-Knowledge Query Planning for an Iterator Implementation of Link Traversal Based Query Execution", in Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P. and Pan, J. (Eds.), *Proceedings of the 8th Extended Semantic Web Conference*, Vol. 6643, Lecture Notes in Computer Science, Springer, pp. 154–169, available at: http://olafhartig.de/files/Hartig_ESWC2011_Preprint.pdf.

[20] Verborgh, R., Vander Sande, M., Colpaert, P., Coppens, S., Mannens, E. and Van de Walle, R. (2014), "Web-Scale Querying through Linked Data Fragments", in Bizer, C., Heath, T., Auer, S. and Berners-Lee, T. (Eds.), *Proceedings of the 7th Workshop on Linked Data on the Web*, Vol. 1184, CEUR Workshop Proceedings, available at: http://ceur-ws.org/Vol-1184/ldow2014_paper_04.pdf.

[21] Matteis, L. and Verborgh, R. (2014), "Hosting Queryable and Highly Available Linked Data for Free", in *Proceedings of the ISWC Developers Workshop 2014*, Vol. 1268, CEUR Workshop Proceedings, pp. 13–18, available at: http://ceur-ws.org/Vol-1268/paper3.pdf.

[22] Rietveld, L., Verborgh, R., Beek, W., Vander Sande, M. and Schlobach, S. (2015), "Linked Data-as-a-Service: The Semantic Web Redeployed", in Gandon, F., Sabou, M., Sack, H., d'Amato, C., Cudré-Mauroux, P. and Zimmermann, A. (Eds.), *The Semantic Web. Latest Advances and New Domains*, Vol. 9088, Lecture Notes in Computer Science, Springer, pp. 471–487, available at: http://linkeddatafragments.org/publications/eswc2015-lodl.pdf.

Cite this article in your publications

- Use the **BibTeX entry** to easily refer to this article.
- Alternatively, you can refer to this article as:

Verborgh, R. (2017), “Piecing the puzzle – Self-publishing queryable research data on the Web”, in Auer, S., Berners-Lee, T., Bizer, C., Capadisli, S., Heath, T., Janowicz, K. and Lehmann, J. (Eds.), *Proceedings of the 10th Workshop on Linked Data on the Web*, Vol. 1809, CEUR Workshop Proceedings.

Comment on this article

8 Comments | Ruben Verborgh | 1 Login

Recommend | Share | Sort by Oldest

Join the discussion…

LDOW2017 Reviewer 1 • 3 months ago

The author proposes a publishing pipeline for personal research data. The pipeline involves automated extraction of RDF triples from a personal website, using reasoning to complete the data (primarily to support access via different vocabularies), and publishing the data accessible via Triple Pattern Fragments. Use cases and requirements are well described, results also include performance and quality measurements.

Also on the positive side: The pipeline is made available in open source for other to easily adopt.

The submission is well written and easy to read. The topic is highly relevant for the workshop and worth to be presented and discussed at this venue.

One critical point: The approach is positioned in contrast to centralized research portals. However, it is not really discussed how an ecosystem of publishing and consuming research data in a decentralized manner would work on a larger scale. On the publishing side: the author – who is a researcher on the topic of publishing Linked Data – has shown that the pipeline works for him, but he is not really representative for the average researcher. The author also leaves out questions how applications consuming this decentralized data could be built effectively. I would expect such kind of discussion in the section on open questions.

Reply • Share ›

Ruben Verborgh Owner ↗ LDOW2017 Reviewer 1 • 2 months ago

Dear reviewer,

Thanks for your comments.

As per your suggestion, I have updated the text to explicitly mention the development of a large-scale ecosystem and an efficient application layer as open questions. I have emphasized the needed steps to make the pipeline work on other’s websites. To address those without knowledge of RDFa, the conclusion now hints at annotation tools and editors.

Best regards,

Ruben

Reply • Share ›

LDOW2017 Reviewer 2 • 3 months ago

The paper presents an approach to query the author's publications, including reasoning to integrate the data and provide better recall on the query answers. The paper presents the scenario involving the author's publications and contrasts TPF with a link-traversal approach. The idea of using reasoning in conjunction with query processing is timely. However, the paper completely ignores the state of the art and related work in the area. While the text reads well, it's more like a blog post than a scientific paper. Hence, I recommend a reject.

Reply • Share ›

Ruben Verborgh Owner → LDOW2017 Reviewer 2 • 2 months ago

Dear reviewer,

Thanks for your comments.

Please note that the approach is not limited to my publications but applicable to RDF on websites in general. I also provide a strategy—and the source code—to apply this to other websites.

Regarding the state of the art, I acknowledge that this submission does not have a dedicated “related work” section. However, there are several references and links to the state of the art in the text itself. Thanks to the pipeline introduced in this article, you can even query its references to existing works.

Despite its inclusion on my website, this submission is a scientific paper and not a blog post, as evidenced by its methodology, experimental setup, and general scientific approach to the description of the proposed pipeline. My blog posts follow a different style and concept than my scientific publications.

Best regards,

Ruben

Reply • Share ›

Sarven Capadisli → LDOW2017 Reviewer 2 • 2 months ago

I find the claim "blog post than a scientific paper" rather disingenuous. What elements of a "scientific paper" do you consistently find at LDOW (or similar calls) that was missing in this *article*? Or did the use of first-person pronoun, instead of the royal "we" had some impact?

What constitutes a "blog post"? It'd be useful to know the differences you had in mind.

Reply • Share ›

Ruben Verborgh Owner → Sarven Capadisli • 2 months ago

I don't suspect disingenuity here, but the end result is the same: the claim isn't precise enough. The reviewer has drawn a conclusion, which might or might not be correct. However, the arguments are not presented, yet these would be what I need to refute and/or fix it.

Doing the same thing, I'd say that the reviewer left a comment rather than a review ;-)

Having revisiting my own article again, I think the main thing that might give it a feeling of a blog post is its use of familiar and direct language. I paid attention to not get unnecessarily complicated, as is good (but unfortunately rare) practice,

which might give the feeling that the content is too simple and/or unscientific. This is a problem of association rather than a quality problem: many scientific articles use (unnecessarily) difficult language, while many blog posts use straightforward language. In that sense, “difficult language” might be an (unreliable) predictor for scientific quality, as there is a correlation, but not a causal relationship. Had the text sounded “more scientific” (i.e., more complicated), the feeling might have been different, but also the audience and reach—which is ultimately the most important thing.

Reply • Share ›

LDOW2017 Reviewer 3 • 3 months ago

The paper reports on a quite exciting experiment: an ETL pipeline that can be applied on a personal web page in order to produce a queryable set of triples with up to date information about a researcher's publications and other activities. The reasoning component allows completing the triple set with respect to the commonly used ontologies, so that this burden is off the annotators (the researcher).

The discussion points that I was missing are
- what one needs to do to apply the approach,
- how this approach may help the rest 99% of researchers who do not have proper annotations or even their personal web pages

Reply • Share ›

Ruben Verborgh Owner → LDOW2017 Reviewer 3 • 2 months ago

Dear reviewer,

Thanks for your comments.

The needs for applying the approach are now better emphasized.

This article indeed starts from the presence of RDFa and is in that sense not directed at the 99% who lack knowledge to create such markup. However, the pipeline’s capabilities provide an additional motivation for those who have the knowledge but currently do not publish RDFa. I updated the conclusion to hint at annotation tools and editors for the 99%, and I differentiate between *production* and (direct) *publication* of RDFa (not in scope of the article) versus *queryability* and *consumption* (in scope).

Best regards,

Ruben

Reply • Share ›